(NASA-CR-190209) LV SOFTWARE SUPPORT FOR SUPERSONIC FLOW ANALYSIS Semiannual Technical Report, 15 Oct. 1991 - 14 Apr. 1992 (Georgia Tech Research Inst.) 12 p
N93-11074
Unclas
G3/34 0083861

LV Software Support for Supersonic Flow Analysis

Semiannual Technical Report
October 1991 to April 1992

for
NASA Grant NAG3-1215
Mr. Royce D. Moore, Technical Officer
Turbomachinery Branch, Mail Stop 77/6
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

by
Dr. William A. Bell, Project Director
Georgia Tech Research Institute
SIgnature Technology Laboratory
Atlanta, GA 30332

Contracting through
Georgia Tech Research Corporation
Centennial Research Building
Georgia Institute of Technology
Atlanta, GA 30332

Progress from October 1991 to April 1992

Aerospace propulsion systems involve some of the most complex flow fields known. Performing the detailed diagnostics for research and analysis requires advanced instrumentation such as the laser velocimeter (LV) to probe the high-speed, unsteady flows. During the first half of this report period, the NASA Lewis Research Center (LeRC) found that existing LV counter processor technology proved inadequate for measuring the flow fields involved. Consequently, a new, high-speed, correlation processor was tested and shown to yield accurate velocity data under anticipated flow conditions. However, this new instrument requires modification of the existing software suite developed under this grant. The remainder of this year will be devoted to this modification of the data analysis software to process the data files generated by the correlation processor.

The progress made last year under this grant led to the acceptance of the attached paper for presentation at the AIAA 17th Ground Testing Conference in Nashville, Tennessee July 6-8, 1992. The progress during this year concentrated on transferring the data through the TELNET network for data analysis and display. Using either the DISSPLA or NASADIG graphics libraries, techniques were developed for displaying the graphics on a wide variety of terminals online. For copies of the displays, the operator can use either the terminal's copying capability, if available, or he can convert the graph to a postscript file and obtain a copy on a compatible laser printer. Both capabilities were tested using the TELNET network to transfer the graphics commands to a remote terminal for on-line display or to a laser printer for off-line copies of report quality.

With the selection of the TSI IFA correlator for LV signal processing, the existing data analysis software required modification to handle the formats unique to the correlator. TSI has sent the documentation of data formats from the IFA system, so modification of the data analysis software can begin. The remainder of this year will be devoted to the following tasks:

1. modify the existing data analysis software to accommodate the formats of the new LV processor;
2. test the existing data reduction routines using simulated LV data;
3. perform final operational testing;
4. submit the final report along with updated user's and programmer's manuals.

LV Software Support for Supersonic Flow Analysis

W. A. Bell[1]
Signature Technology Laboratory
Georgia Tech Research Institute, Atlanta, GA 30332
and
J. Lepicovsky[2]
Sverdrup Technology, NASA LeRC Group
2001 Aerospace Parkway, Brook Park, OH 44142

Abstract

The software for configuring an LV counter processor system has been developed using structured design. The LV system includes up to three counter processors and a rotary encoder. The software for configuring and testing the LV system has been developed, tested, and included in an overall software package for data acquisition, analysis, and reduction. Error handling routines respond to both operator and instrument errors which often arise in the course of measuring complex, high-speed flows. The use of networking capabilities greatly facilitates the software development process by allowing software development and testing from a remote site. In addition, high-speed transfers allow graphics files or commands to provide viewing of the data from a remote site. Further advances in data analysis require corresponding advances in procedures for statistical and time series analysis of nonuniformly sampled data.

Introduction

The next generation of aircraft designs continue to involve increasingly complex, high-speed flows. Performing the detailed flow diagnostics to properly evaluate these designs requires advanced instrumentation applicable to complex, high-speed, unsteady flows. A laser velocimeter (LV) system is such an instrument. Since the LV is a proven technique providing nonintrusive measurements, it is capable of acquiring the velocity field with minimal interference to the flow.

However, these complex, high-speed flow fields push the limits of present analytical, numerical, and experimental techniques for fluid dynamics. The LV is no exception. To ensure the proper operation of the instrument in the presence of these flows, the system must be configured, calibrated, and tested on line to ensure accurate results under conditions near the operating limits of the LV. The measured velocity data must be rapidly acquired and presented to the operator on line so that he can respond to changing test conditions. The data must be further processed off line to perform more advanced flow diagnostics such as time series analysis, velocity bias correction, and computation of higher order statistics.

Based on the flow conditions for the application of interest in this paper, analysis of the laser velocimeter system indicated short particle residence times within the measurement volume. For instance, as a particle traverses the measurement volume at a speed V, the total traverse time T across the volume diameter D is D/V. For a diameter of 200 micrometers and a velocity of 500 meters per second, T is 0.4 microseconds. For an LV frequency processor sampling at 400 megahertz, only 100 samples would be taken for analysis by an extended the fast Fourier transform (FFT) technique. With a counter processor at a one gigahertz clock frequency, the predicted velocity resolution could also yield reesults comparable to the FFT processor for the present application. Therefore, the software design allows for either possibility.

The LV system is part of an overall instrumentation suite for advanced flow research, which includes positioning and monitoring instrumentation. Consequently, the LV software must be capable of being integrated into an existing computer system for on-line data acquisition, processing, and presentation. To meet this requirement, a modular design approach is employed, which enhances cohesion but suppresses coupling with external instrumentation software.[1,2]

. 1. Senior Research Engineer, Senior Member, AIAA

. 2. Senior Supervisor, Associate Fellow, AIAA

The primary objectives of this or any other software package for aerodynamic testing are to acquire, analyze, and present experimental data in a form that makes sense to the operator. This paper describes the software developed that allows the operator to configure and checkout the LV system prior to and during a run. This setup procedure establishes the operating conditions for the LV interface in response to the particular flow conditions, and incorporates a rotating machinery resolver to provide timing synchronization for conditional sampling. In addition to initializing the instruments, the software package provides a means of specifying LV calibration constants, controlling the sampling process, and identifying the test parameters. A network link established using the TELNET protocol provides access to the computers at the test site from the remote development site. Thus, the software can be developed and tested at the development site using the network to access the equipment at the test site. This procedure minimizes turnaround time, which increases the responsiveness to the changing needs of advanced flow research.

With the basis established for controlling the operation of the LV system, the next phase of software development will concentrate on the analysis and presentation of the LV data. Because of the diversity of the types of flows to be investigated, the data analysis will include velocity statistics, time series analysis, and conditional sampling. Existing methods described in Refs. 3 through 7 can be modified to apply to any particular LV system. In addition, the LV is part of an overall instrumentation system for acquiring and analyzing complex, unsteady flow fields of propulsion systems. Consequently, the software design must accommodate the total environment and address integration and coordination issues.

LV Software Development

The approach to software development for the LV system follows the principles developed in Refs. 1 and 2. These principles support a modular, structured software development process. Since the LV is part of an overall instrumentation suite, these principles facilitate the integration of the LV software into the overall test support software.

Recognizing the need for adequate documentation, the software development established a set of guidelines in documenting the requirements, design, code, testing, and operation. To avoid the voluminous reports often generated to satisfy documentation requirements, a structured, object-oriented approach was used in the development of the current LV software. With this approach the documentation and the software can be broken down numerous small descriptions of the various modules that comprise the overall software package. This allows updates or modifications to be restricted to small portions of the manual and prevents a major rewrite for each software revision.

Extended versions of FORTRAN-77 support the structured approach. The two major features offered are the STRUCTURE statement for defining and strongly typing variables and the expansion in the size of subroutine and variable names from six to 32 characters. These features allow subroutine names to be more descriptive (CONDUCT_TEST rather than GDATA) and allows more lucid, structured variable naming conventions (LV.VELOCITY.SAMPLES rather than NUM).

For the current software, structured analysis consists of taking the three major functions - setup, data acquisition, and data analysis - and breaking them up into subfunctions, as shown schematically in Fig. 1. This process is repeated until the software tasks are partitioned into the smallest possible components. For instance, the setup function can be decomposed into defining the test, configuring the LV, and defining the traverse sequence. Configuring the LV consists of defining the conversion constants, setting up the LV electronics, and establishing the statistical environment, and so on. Data flow diagrams provide the means for keeping track of the decomposition and allow a means of defining subroutine names (for instance, SET_LV_INTERFACE for the subroutine that performs the function that defines the LV interface name and status).

At each stage of the functional decomposition the variables involved are defined in a data dictionary. Using the STRUCTURE statement in extended FORTRAN (or the RECORD statement in Ada) is a convenient way of keeping track of these variables. It also leads to more recognizable variable names. Using the data flow diagrams along with the data dictionary allows the software developer to code the requirements in FORTRAN using the STRUCTURE statements for variable definition, the abbreviated subfunction definition for the subroutine name, and comments describing the function of the subroutine in more detail. This approach minimizes software development time, allows checking of the preliminary code from the requirements to the operational phases, and gives the programmer a way to code immediately in a structured manner, which is what many programmers do anyway, but without any overall plan.

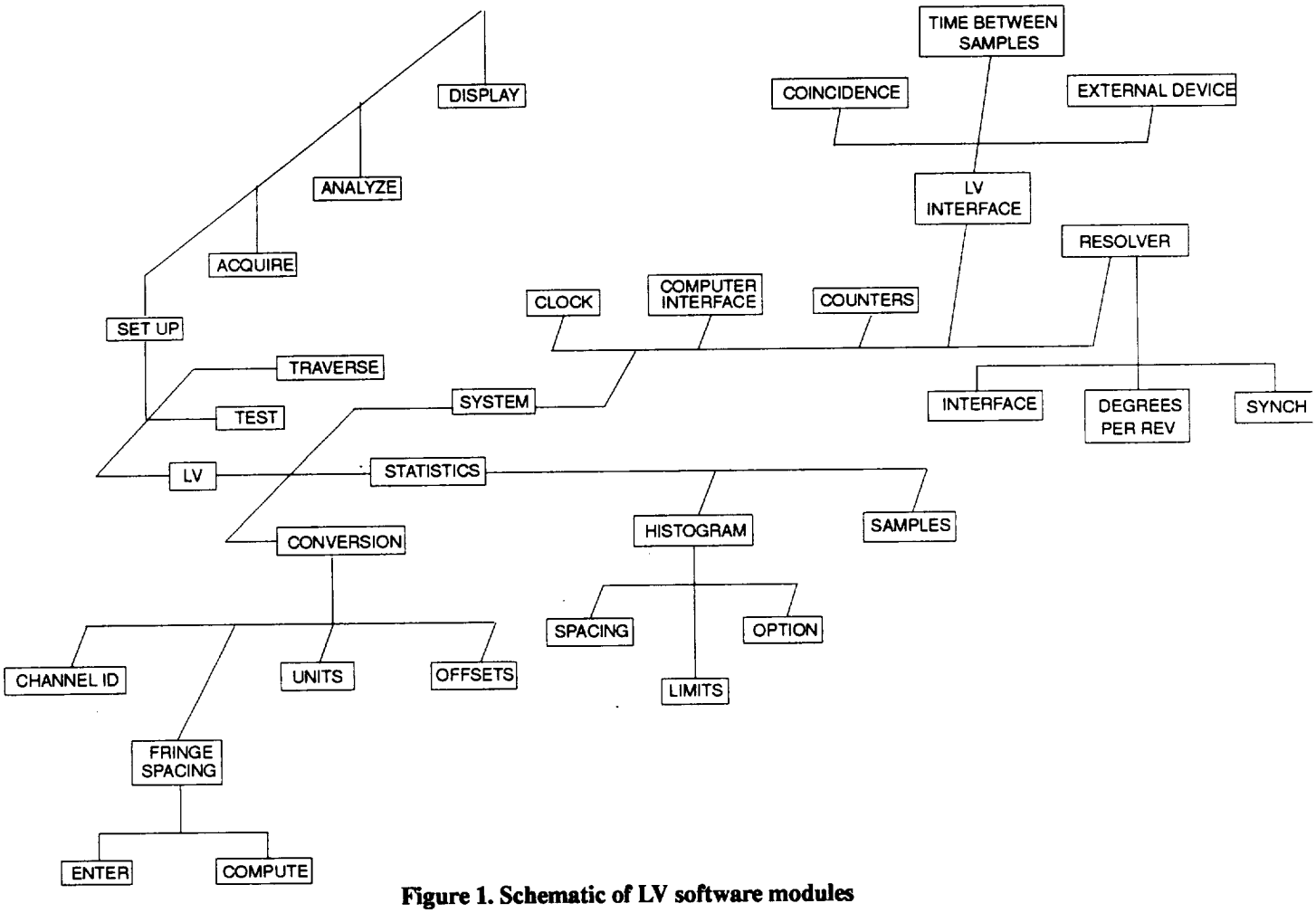

Figure 1. Schematic of LV software modules

For most wind-tunnel applications, the LV velocity data can be presented in three forms, as shown in Fig. 2: instrument readings, physical quantities, and graphics. Initially, the data consists of instrument readings which may or may not be in the form of a physical quantity, such as position or velocity. This form is most convenient for a technician or operator to troubleshoot the instrument. The current LV software provides routines which acquire and present the raw instrument readings to facilitate instrument diagnostics during the setup procedure. For monitoring test conditions during the run, the software converts the instrument readings to physical quantities so that the operator can ensure proper test conditions during on-line acquisition. Finally, during off-line analysis, the data sets are combined to produce a composite graph of the data. This form allows conveyance of the maximum amount of information in the minimum amount of time.

The operator interface relies heavily on screen management utilities that support multiple windows for command entries and menus describing the function of each command. Making use of the screen management facility allows the operator to concentrate on the test rather than memorize commands by providing a more user-friendly interface. To maintain independence of the software on a particular terminal, the screen manager used supports a wide variety of terminals from a generic ASCII terminal to the sophistication of a workstation.

Software in support of on-line testing performs three main functions: acquisition, analysis, and presentation. The acquisition process consists of identifying the test, configuring the instrument and initiating data transfers. The software development described in this paper concentrates on configuring the instrument prior to and during acquisition. The software stores the current test and

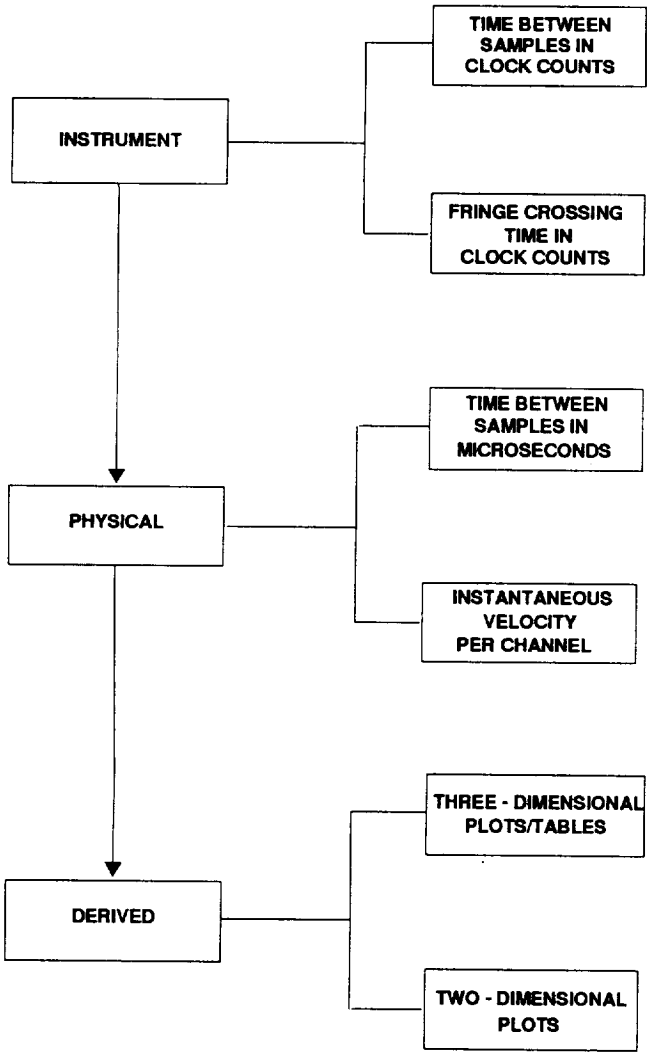

Figure 2. Forms of LV data

instrumentation configurations for retrieval, saving the operator from having to re-enter the information after exiting the program. The variables for test configuration are grouped according to expected frequency of access and modification and are stored in configuration files separated according to function.

Data analysis and presentation depend on the needs of the end user. For instrument diagnosis and testing, which is important during checkout prior to testing, the data analysis and presentation consists of printing out the raw instrument readings for visual inspection. For analyzing flow data after a run, the data analysis involves converting the instrument readings to physical quantities, such as velocity and time between samples for the LV, and presenting summary statistics for on-line test monitoring and control, as shown in Fig. 3.

Because of the nature of the testing environment, it is important that the software be robust. In the present context, robustness means that, at a minimum, the software responds to errors induced by the instrument or operator by issuing an error message to the monitor and returning to a known condition. Using error handling routines that are part of the operating system, the software attempts to identify the error source, type, and correction. Whenever possible, error recovery is attempted.

Prior to software development, the software and hardware requirements definition led to the use of the existing computers using the VMS operating system. VMS supports the real-time environment of advanced flow research through sophisticated system services for control of data transfers and error detection and recovery. Following the procedures implemented at the test site, low-speed data transfers required for instrumentation setup employ the RS-232 serial interface standard. The high-speed data transfers from the LV interface take place across a high-speed parallel interface, which uses direct memory access (DMA) to transfer the raw data to the on-line processor.

The modular design of the LV data acquisition software allows the components to operate separately for independent configuration and testing or concurrently during a run. This design also facilitates integration with other software packages and instrumentation systems as the need arises.

Software testing consists of three parts - initial, simulation, and operational. Initial testing consists primarily of finding and correcting errors in the coding or linking of the software. The next level of testing is simulation, where the data transfers involve a local device, such as a terminal, that simulates the operation of the instrument under test. The commands sent to and the response from the device can be tested using this procedure. The performance of the software under both standard and anomalous conditions can be ascertained. Finally, operational testing is performed using the actual instruments under standard operating conditions.

The errors handled by this program are of two types - operator and instrument. Operator errors handled consist of erroneous commands and data entries. Erroneous commands result in the issuance of the appropriate

RUN	CODE	AXIS	POSITION	CHANNEL	SAMPLES	MEANS	STD
1	1	x	0.000	1	100	31.2	1.1
		y	0.000	2	100	0.6	1.1
		z	0.000	3	100	0.5	1.0
1	2	x	0.000	1	100	31.2	1.1
		y	0.000	2	100	0.7	0.9
		z	0.000	3	100	0.8	0.8
1	3	x	0.000	1	100	30.1	0.9
		y	0.000	2	100	1.1	0.9
		z	0.000	3	100	0.4	1.0

Figure 3. On line summary statistics

prompt with no action being taken. Erroneous data entries cause the current values of the variables to be retained.

Instrument errors result in message consisting of three parts. The first is the error source. The second part described the error condition. The third part gives a procedure for recovering from the error. An example of an error message is given in Fig. 4 resulting from entering an invalid name for the LV multichannel interface. This particular message results in the interface status being set to OFF to prevent possible problems in subsequent calls to the device.

During instrument operation, the most common error arises from the time-out condition. This condition occurs when the time for the instrument to respond to a command exceeds a maximum value set in the software for the particular device. Because this error can arise from a variety of sources, all software for controlling an instrument is designed to handle this condition and return the operating status to a known state.

In order to conduct final, operational testing at the onsite test facilities, the software must be transported from the development site to the target computer. An interface to the onsite computing network was established using TELNET. Not only does this arrangement allow transfer of the software between the development and application facilities, but it also provides a means for operating the software from a remote site. This capability was applied to transfer, compile, and test the LV software on the processor at the test facility. The final operational testing of the software occurred upon completion of the final installation and checkout of the LV hardware. More recently, graphics routines have been written using the graphics libraries at the test facility to generate text files consisting of graphics commands. These files can then be transferred using standard FTP file transfer protocol to a remote site for display on a laser printer.

Results

A formal software development cycle consists of four parts - requirements definition, design, coding, and testing. During the requirements definition phase of the LV software development, the final hardware and software systems were selected and procured. The hardware at the test site consists of the LV interface to the computer, an internal LV bus to the counter processors, and a rotating machinery resolver interfaced to the on-line computer. Since the computer at the software development facility maintains compatibility with the processor at the test site, software written at the development site transports to the target computer without conversion to another operating system or compiler. This greatly facilitates software development. After reviewing the manuals for the LV and rotating machinery resolver, the

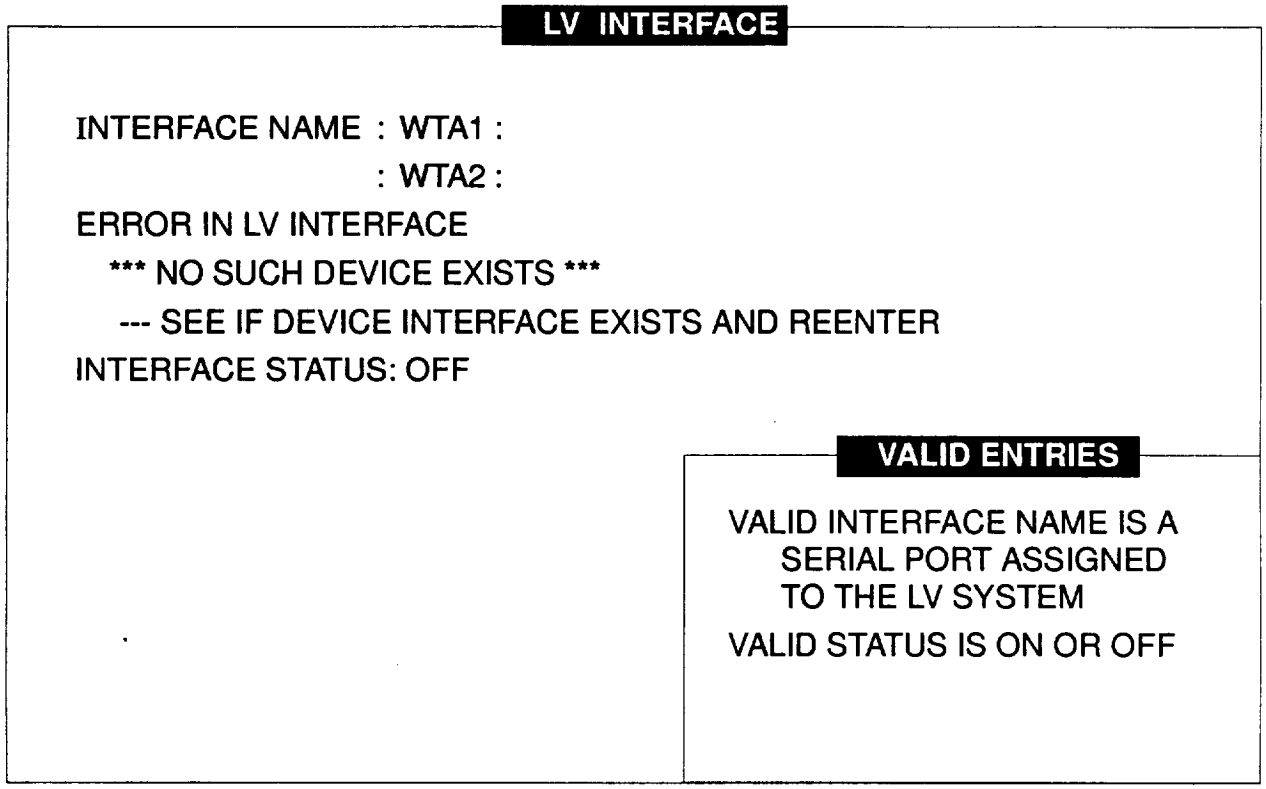

Figure 4. Example of an error message

commands and techniques for controlling the instruments were established and the software requirements defined using structured specification.[2]

Upon program execution the menu shown in Fig. 5 appears on the screen. The commands currently supported are SE (SEtup), RU (RUn), ST (Simulate a Test) and EN (ENd). AN (ANalyze) and RR (Run and Reduce) are included for future upgrades upon final requirements definition, and are presently expected to be modifications of existing software packages developed in Refs. 3 through 7. SE allows modification of the test and instrument parameters; whereas RU sets the position and acquires, analyzes, and displays the LV velocity data. ST simulates a test using the LV system and allows analysis of the instrument resolution and software checkout.

Following the coding of the routines for identifying and controlling the test, the software for configuring the LV interface was coded and verified using a terminal to simulate the response of the interface to operator commands. In addition to fringe crossing times for determining velocity, the LV interface provides the time between samples required for the removal of velocity bias[7] and time series analysis[6] of the unsteady velocity data. The coincidence between channels can be set to support the measurement of cross correlations of the velocity data between channels. The software for performing statistical and time series analysis of the velocity data is based on the algorithms in Refs. 3 through 7.

The rotating machinery resolver allows the velocity sampling by the LV to be synchronized with the rotation of a turbine shaft or with a timing pulse. This synchronization allows conditional sampling of the velocity, which has a variety of applications in studies of aeropropulsion and aeroacoustics.[5] Conditional sampling is a basic technique for synchronizing the sampling process with periodic disturbances. The resolver provides the capability for setting up the sampling procedure. The conditional sampling techniques are given in Ref. 5 and software modules developed are shown in Fig. 1 under the resolver. To increase robustness, the software checks for errors in the operator entries to minimize disruption from erroneous inputs.

The setup function provides for configuration of the LV, test description, and traverse sequence. The traverse option is included to step through a positioning sequence using a linear positioning system in order to obtain spatial distributions of velocity and vorticity information. Throughout the setup procedure, entering a carriage return or blanks retains the displayed values.

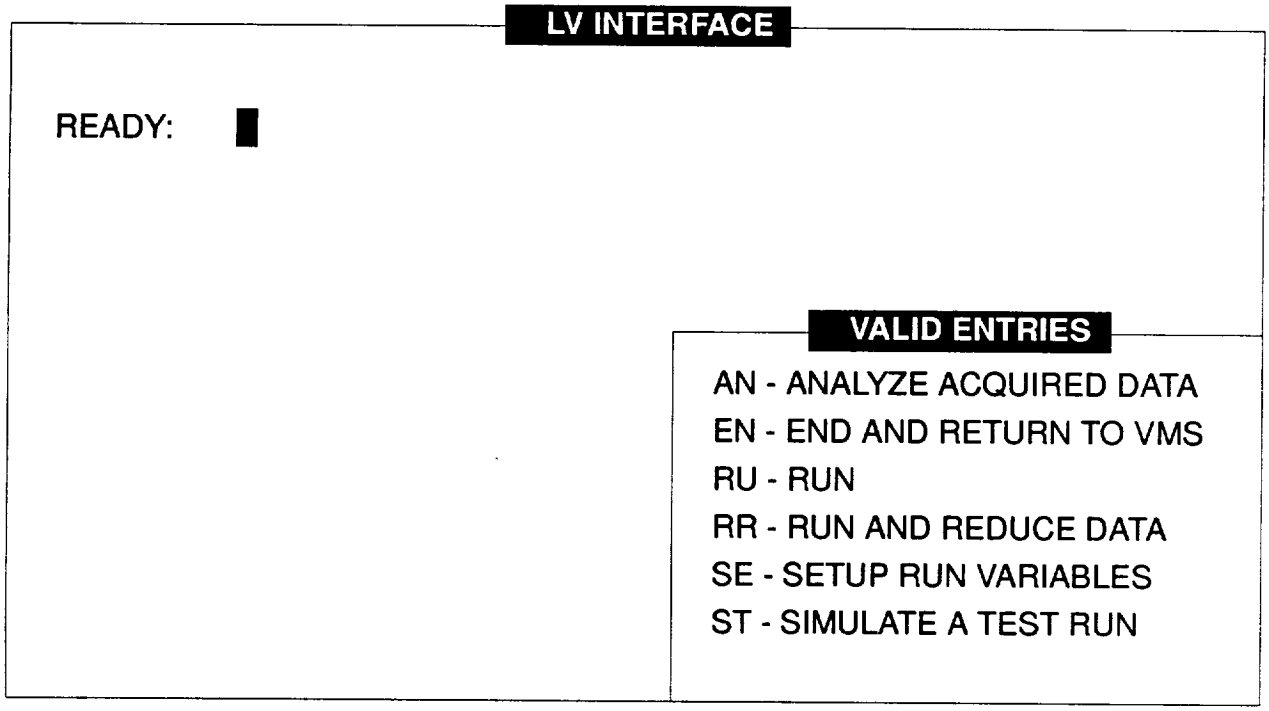

Figure 5. Main menu

Invalid entries result in again displaying the current values for reentry.

Setting up the LV involves the menu shown in Fig. 6. Here, the commands necessary to convert the raw LV data into appropriate velocity units, to define the LV interface configuration, and to control the velocity statistics can be entered.

To set the conversion constants, the cursor is positioned under the desired channel or channels and a valid identification for the velocity component is entered. For the three-dimensional LV configuration, entering a nonzero angle flags the data reduction routine that three-dimensional data are available and additional processing must be applied to resolve the third velocity component. Invalid entries result in the current settings being displayed again for reentry. After channel identification, the operator can enter the beam spacing directly, or allow the software to compute the fringe spacing from values of beam spacing, focal length, and wavelength. The velocity can be expressed in either English or SI units. The final entries required for converting raw LV time words into velocity data are the velocity offsets for each channel produced by the Bragg shift.

Defining the LV system configuration consists of setting the clock frequency, configuring the computer interface, defining the number of channels, and setting up the operation of the LV, as shown in Fig. 1. The clock frequency of current systems is one gigahertz and determines the resolution of the fringe crossing time. The computer interface command provides a means of defining the LV computer interface and status. A valid interface name is assigned by the operating system to the high-speed interface between the computer and the LV. If an error condition occurs during setup, the interface status is turned off and the operator notified.

Prior to testing, the multichannel LV bus must be configured for the number of channels of counter processor data. Entering a valid channel number between one and three sets the number of channels and the software then forms and sends the command to the LV controller. If an error occurs in the transmission, then message is sent to the screen to notify the operator.

Setting the operating mode of the LV provides the means for tailoring the multichannel interface configuration for a particular run and consists of controlling the coincidence, input channels, sampling mode, and sampling times. In many applications, such as shear stress computation, all LV channels must contain valid data in

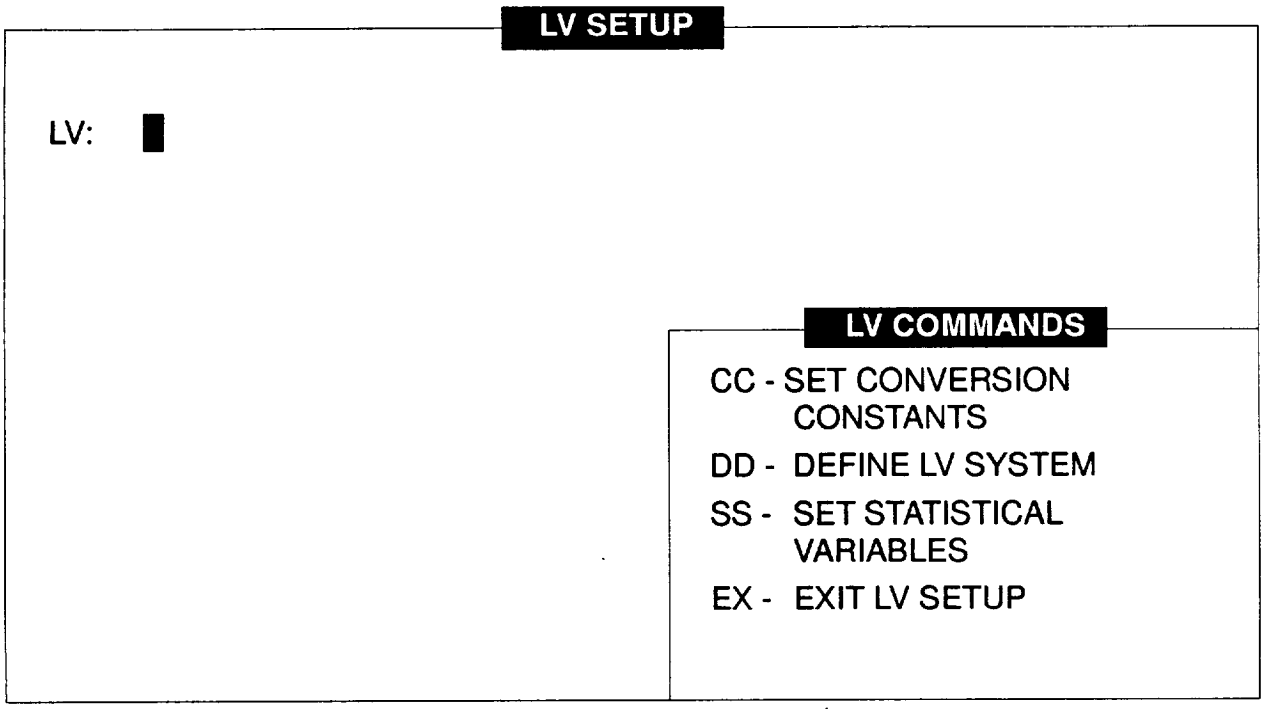

Figure 6. LV setup menu

order to be accepted. This coincidence must occur within a specified time interval controlled by the operator or it may be disabled by the operator. In addition to fringe crossing times and the time between samples, the LV controller allows from zero to 16 additional inputs from various devices. Normally, the LV acquires data whenever a particle enters the measurement volume, which tends to occur at nonuniform time intervals. The LV can be configured to sample at uniformly spaced time intervals between one microsecond and one second. Of course, the time between particle arrivals should be much greater than the sampling interval. The sending of time between samples can be enabled or inhibited. If the time between samples are inhibited, the time between data and even time sampling statuses are flagged as off.

For synchronizing flows with external time marks, a timing mark is provided using a rotating machinery resolver. After setting the computer interface to the device, the resolver can be set to acquire velocity data at specified angular intervals. Not only can the sampling intervals be controlled, but the resolver allows the error tolerances in angular resolution to be set to one of four tolerances. Additional commands set the number of degrees per cycle to 360 or 720 degrees per cycle.

To control the LV velocity statistics, the operator can set the number of samples, confidence limit for histogram clipping, histogram spacing, and the option to turn histogram online displays on or off. To enter the parameters describing the test, the operator can set the test number, tunnel, run number, code number, run title, and reference distances, velocities, and angles.

While in the executive menu, entering the command to run initiates the run sequence. The screen display during a run is shown in Fig. 7. The information consists of test information (run and code), position (axis, coordinate), and velocity statistics (channel number, samples, means, and standard deviations). A menu overlays the display as shown in Fig. 7 to set the position. Valid entries consist of an axis and the desired coordinate value for that axis. The values are then updated. Entering a carriage return instead of an axis removes this display and initiates the run. Entering S for the axis terminates the run sequence, clears the screen, and returns control to the executive menu.

Entering the command ST (Simulate a Test) at the executive menu display results in a run sequence using simulated data instead of actual velocity data from the LV system. The velocity data are generated at an average sampling rate of 1000 samples per second with the time

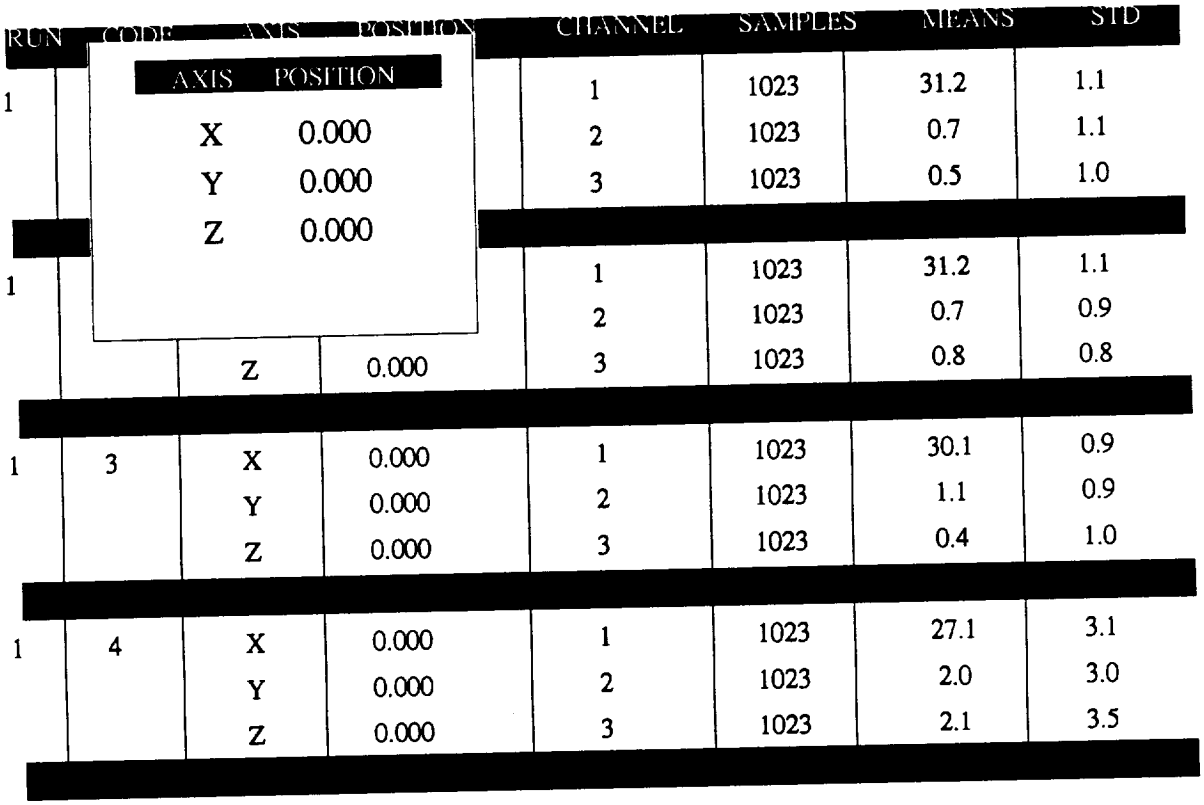

Figure 7. On-line display

between samples distributed according to a Poisson distribution. Currently, this capability supports software development. Future enhancements can include generation of simulated data allowing the operator to control the amplitude and frequency of the velocity fluctuations for checkout of advanced data reduction algorithms and instrument resolution.

Data analysis routines have been written based on the algorithms in Refs. 3 to 6. In general, the LV presents two difficulties in statistical and time series analysis of the measured velocities. The first is the dependence of the sampling on particle velocity, which leads to a bias in the computed statistics. The method developed in Ref. 7 is used to correct for the bias, if desired, since this technique allows the degree of coupling between the velocity and sampling statistics to be determined. The second difficulty arises from the random sampling resulting from the Poisson distribution of the time between particle arrivals in the sampling volume. This nonuniform distribution of sampling intervals produces different samples along a waveform, even if the waveform is periodic. This nonuniform sampling increases the variance of estimates of the correlation and power spectra of the signal, even for advanced processing techniques.[8]

As shown in Refs. 3-6, the data can be presented in a variety of forms in both two- and three-dimensional graphics. The present uses off-the-shelf graphics packages such as NASA-DIG and DISSPLA to generate the graphics, although newer libraries such as PV-WAVE are specifically designed for data analysis and presentation applications. Since the graphics are generally highly dependent on the particular requirements of the operator, the graphics routines are kept as independent as possible from the rest of the software to allow for rapid, flexible modification of the routines. To achieve as much independence as possible, all plotting programs rely on data files rather than on-line RAM to generate final, report-quality plots.

Summary and Conclusions

Using proven structured programming methods, the software for configuring an LV counter processor system has been developed. With the increasing sophistica-

tion of today's instrumentation, a structured software development approach is highly desirable. The LV system includes up to three counter processors and a rotary encoder. The software for configuring and testing the LV system has been developed and tested and included in an overall software package for data acquisition, analysis, and reduction. Error handling routines have also been incorporated that respond to both operator and instrument errors which often arise in the course of measuring complex, high-speed flows. The use of networking capabilities greatly facilitates the software development process by allowing software development and testing from a remote site. In addition, high-speed transfers allow graphics files or commands to provide viewing of the data from a remote site. Further advances in data analysis require corresponding advances in procedures for statistical and time series analysis of nonuniformly sampled data.

Acknowledgments

This work was performed under Grant NAG3-1215 with NASA LeRC under the direction of Royce Moore, Technical Monitor. The use of the extensive computing resources at Georgia Tech and Lewis Research Center are gratefully acknowledged along with the system management assistance of Glenn Christman.

References

1. Booch, G., *Software Engineering with Ada*, 2nd Edition, Benjamin/Cummings Publishing Company, 1987.
2. deMarco, T., *Structured Analysis and System Specification*, Yourdon, Inc., 1979.
3. Bell, W. A. and Cornelius, K. C., "An Experimental Investigation of a Laminar Separation Bubble on a Natural Laminar Flow Airfoil", Paper AIAA-87-0458, AIAA 25th Aerospace Sciences Meeting, Reno, NV, January 1987.
4. Novak, C. J. and Huie, C. R. "3-D Laser Velocimeter Investigations of a Generic Fighter Flowfield", Paper AIAA-87-0331, AIAA 25th Aerospace Sciences Meeting, Reno, NV, January 1987.
5. Lepicovsky, J., Bell, W. A. and Ahuja, K. K., "Conditional Sampling with a Laser Velocimeter and Its Application for Large-Scale Turbulent Structure Measurement". LG83ER0001, Lockheed-Georgia Company, January 1983.
6. Bell, W. A., "Spectral Analysis of Laser Velocimeter Data with the Slotted Correlation Method", Paper AIAA-86-1102, AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics, and Lasers Conference, Atlanta, Georgia, May 1986.
7. Edwards, R. V. and Meyers, J. F., "An Overview of Particle Sampling Bias", *Second International Symposium on Applications of Laser Anemometry to Fluid Mechanics*, Lisbon, Portugal, July 1984.
8. Mallat, S. G., "Multifrequency Channel Decomposition of Images and Wavelet Models", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No 12, 1989, pp. 2091-2110.